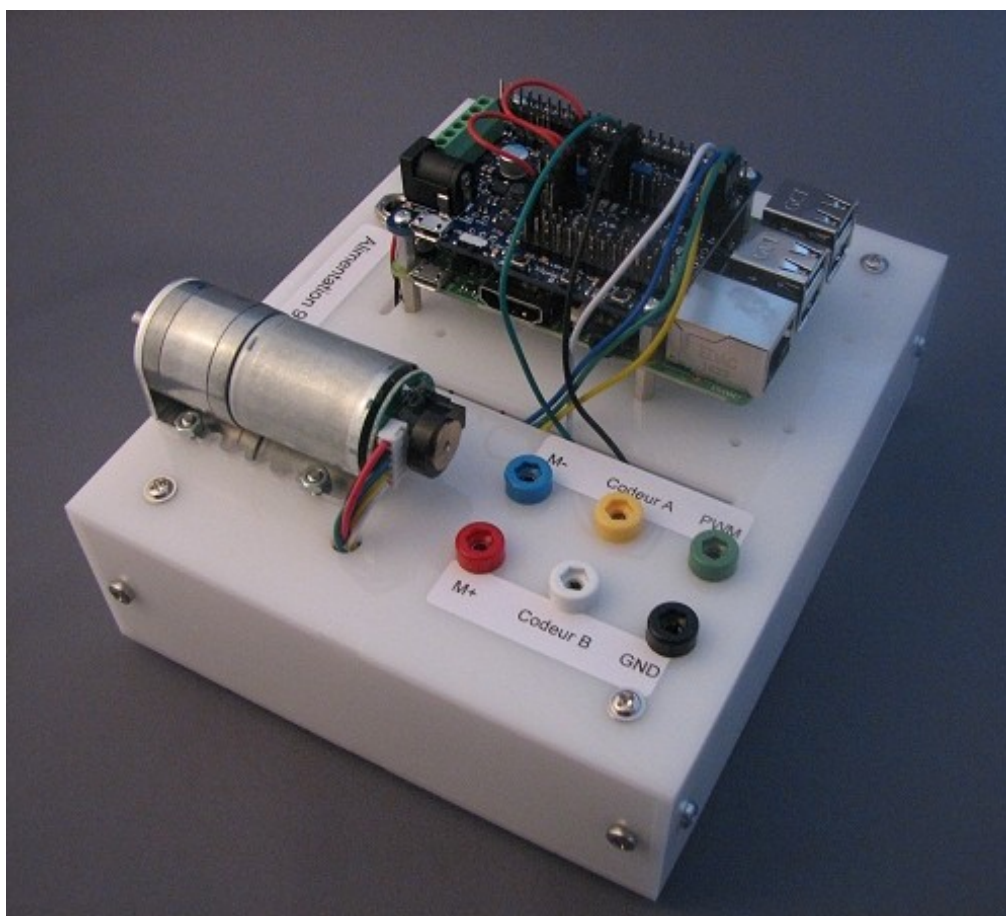




ENSEMBLE « COMMANDE DE MOTEUR À COURANT CONTINU » RASPBERRY C / PYTHON

DOCUMENTATION COMPLÈTE



Date de dernière mise à jour : 02/12/2016

Table des matières

1 - Introduction.....	<u>3</u>
2 - Matériel inclus.....	<u>3</u>
3 - Conformité.....	<u>3</u>
4 - Architecture du système.....	<u>4</u>
5 - Installation de l'IDE Arduino.....	<u>5</u>
5.1 - Installation principale.....	<u>5</u>
5.2 - Installation du support de la carte A-Star.....	<u>5</u>
5.3 - Installation de la bibliothèque complémentaire FlexiTimer2.....	<u>6</u>
5.4 - Téléchargement d'un programme sur la carte A-Star.....	<u>6</u>
6 - Mise en œuvre de l'ensemble.....	<u>7</u>
6.1 - Opérations préalables.....	<u>7</u>
6.2 - Utilisation standard.....	<u>7</u>
6.3 - Précautions d'emploi.....	<u>7</u>
6.3.1 - Connexions d'alimentation sur la carte A-Star.....	<u>7</u>
6.3.2 - Tension d'alimentation.....	<u>8</u>
6.3.3 - Utilisation.....	<u>8</u>
7 - Expériences de commande de moteur à courant continu.....	<u>9</u>
7.1 - Caractéristiques du moteur à courant continu avec codeur incrémental associé.....	<u>9</u>
7.1.1 - Calcul de la vitesse de rotation du moteur.....	<u>11</u>
7.1.2 - Comptage du nombre d'impulsions.....	<u>12</u>
7.2 - Commande en tension avec la carte Raspberry Pi.....	<u>13</u>
7.3 - Asservissement de vitesse avec la carte Raspberry Pi.....	<u>16</u>
7.4 - Asservissement de position avec la carte Raspberry Pi.....	<u>20</u>
7.5 - Commande en tension avec la carte A-Star.....	<u>24</u>
7.6 - Asservissement de vitesse avec la carte A-Star.....	<u>27</u>
7.7 - Asservissement de position avec la carte A-Star.....	<u>31</u>
8 - Accès aux programmes Python.....	<u>33</u>
9 - Important.....	<u>34</u>

1 - Introduction

Cet ensemble permet de réaliser différentes expériences sur la base d'un moteur à courant continu avec codeur incrémental associé, commandé par l'association d'une carte Raspberry Pi et d'une carte A-Star (compatible Arduino).

Cette documentation, ainsi que les programmes associés sont téléchargeables sur notre site Web à l'adresse suivante:

<http://www.3sigma.fr/telechargements>

2 - Matériel inclus

Cet ensemble est livré monté et **fonctionnel (testé par nos soins avant la livraison)**. Il est composé des éléments suivants:

- un boîtier en plexiglas blanc, avec connecteur d'alimentation 5.5mm x 2.1mm, bouton marche-arrêt et douilles 2mm pour la mesure à l'oscilloscope de différents signaux.

Repérage des douilles 2mm:

Rouge	Bleu	Jaune	Blanc	Vert	Noir
+ moteur	- moteur	Voie A codeur	Voie B codeur	Signal PWM	Masse

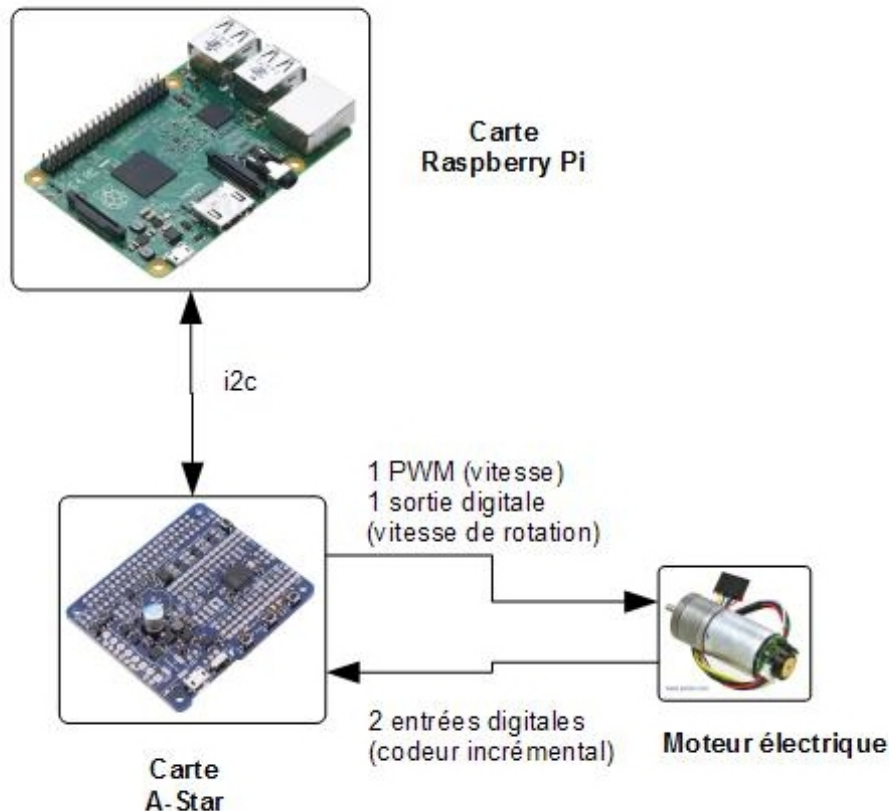
- une carte Raspberry Pi
- une carte A-Star (<http://blog.3sigma.fr/arduino/test-de-la-carte-de-contrôle-robot-a-star-32u4-de-pololu>), compatible Arduino Leonardo
- un moteur à courant continu 6V, rapport de réduction 34:1, avec codeur incrémental 48 CPR. Le moteur est assemblé sur le boîtier par l'intermédiaire d'un support de montage en équerre
- 1 adaptateur USB / Ethernet permettant de communiquer directement entre un ordinateur (connexion USB) et la Raspberry Pi (connexion Ethernet)
- 1 câble Ethernet
- 1 câble micro USB pour la programmation de la carte A-Star
- 1 alimentation 9V, 1A avec connecteur d'alimentation 5.5mm x 2.1mm

3 - Conformité

L'ensemble « Commande de moteur à courant continu », **dans sa configuration livrée aux clients**, est conforme à la directive 1999/EC.

4 - Architecture du système

Le diagramme d'échange de signaux est représenté ci-dessous :



La commande rapprochée des moteurs (comptage des impulsions du codeur incrémental du moteur et génération du PWM de pilotage) est, dans tous les cas, gérée par la carte A-Star, (compatible Arduino) plus performante pour réaliser ces tâches rapides de bas niveau.

Dans le cas où l'on souhaite programmer le système 100 % en code C/C++, les asservissements de position ou de vitesse sont également exécutés par cette carte. Dans ce cas, la Raspberry Pi n'est pas utilisée.

Lorsque l'on souhaite programmer le système en utilisant le langage Python, la carte Raspberry Pi intervient pour exécuter les asservissements de haut niveau. Elle communique avec la carte A-Star via une liaison i2c pour récupérer le nombre d'impulsions du codeur à chaque pas de temps et renvoyer la valeur du rapport cyclique du PWM calculés par les algorithmes de commande exécutés par la Raspberry Pi.

5 - Installation de l'IDE Arduino

Cette opération est nécessaire uniquement si vous souhaitez modifier le programme Arduino exécuté sur la carte A-Star. A la livraison, celle-ci est programmée pour que le système soit utilisable avec des programmes Python exécutés par la carte Raspberry Pi.

Si vous souhaitez commander le système avec des programmes écrits en C/C++ et exécutés sur cette carte compatible Arduino, vous devez tout d'abord réaliser les opérations suivantes.

5.1 - Installation principale

Si ce n'est pas déjà fait, télécharger et installer l'IDE Arduino (<http://arduino.cc/en/Main/Software>).

5.2 - Installation du support de la carte A-Star

Il faut effectuer les opérations suivantes pour ajouter le support de la carte A-Star dans l'IDE Arduino :

- Télécharger et décompresser l'archive suivante dans le répertoire de votre choix : https://www.pololu.com/file/download/a-star-2.0.0.zip?file_id=0J743
- Ouvrir le sous-répertoire « drivers ». Faire un clic bouton droit sur le fichier a-star.inf et sélectionner « Installer »
- Repérer l'emplacement de votre « Répertoire de croquis » en sélectionnant Fichiers → Préférences dans l'IDE Arduino. Ce répertoire est indiqué à l' « Emplacement du carnet de croquis »
- Copier dans ce répertoire de croquis le répertoire nommé « pololu », que vous trouverez dans l'archive précédemment téléchargée
- Télécharger la bibliothèque <http://www.3sigma.fr/telechargements/AStar32U4.zip>
- Décompresser cette bibliothèque dans le répertoire des librairies de votre installation Arduino (typiquement, Documents\Arduino\libraries)
- Redémarrer l'IDE Arduino
- Dans le menu Outils → Type de carte, choisir « Pololu A-Star 32u4 »

5.3 - Installation de la bibliothèque complémentaire FlexiTimer2

Cette bibliothèque permet d'exécuter à cadence fixe une partie du programme Arduino.

Vous pouvez la télécharger à l'adresse suivante: <http://www.3sigma.fr/telechargements/FlexiTimer2.zip>.

Une fois téléchargée, décompressez-la dans le répertoire des bibliothèques de votre installation Arduino (typiquement, Documents\Arduino\libraries) et redémarrez l'IDE Arduino.

5.4 - Téléchargement d'un programme sur la carte A-Star

La procédure à suivre pour la programmation est la suivante:

- Mettre le système sous tension en appuyant sur le bouton marche-arrêt
- Connecter le câble USB reliant l'ordinateur et la carte A-Star
- Lancer le téléchargement

6 - Mise en œuvre de l'ensemble

6.1 - Opérations préalables

Attention : avant la première connexion au système via un câble Ethernet, vous devez installer sur votre ordinateur le pilote de l'adaptateur USB / Ethernet fourni avec le système. Ce pilote se trouve sur le CDROM livré avec l'adaptateur.

6.2 - Utilisation standard

La mise en œuvre de l'ensemble « Commande de moteur à courant continu » est très simple:

- brancher l'alimentation 9V fournie sur le connecteur jack du système
- connecter l'ordinateur à la carte Raspberry Pi en utilisant le câble Ethernet et l'adaptateur USB – Ethernet (USB côté ordinateur)
- commuter l'interrupteur sur la position « I »
- attendre environ une minute le démarrage de la Raspberry Pi
- exécuter l'un des logiciels de pilotage (commande en tension, asservissement en vitesse ou en position, voir plus loin)

6.3 - Précautions d'emploi

Nous insistons sur le fait que cet ensemble est un matériel de développement qui nécessite un certain nombre de précautions d'emploi.

6.3.1 - Connexions d'alimentation sur la carte A-Star

L'ensemble « Commande de moteur à courant continu » étant livré connecté et fonctionnel, il est préférable de ne pas modifier les branchements sur les connecteurs d'alimentation.

6.3.2 - Tension d'alimentation

Cet ensemble est prévu pour fonctionner avec l'alimentation 9V, 1A fournie. Le moteur a une tension nominale de 6V et bien qu'il supporte sans problème des tensions jusqu'à 9V, les programmes disponibles sur la carte Raspberry Pi (ainsi que tous les programmes téléchargeables sur notre site) empêchent que la tension à ses bornes dépasse la valeur de 6V.

Il est possible d'utiliser un autre bloc d'alimentation à condition de bien respecter les points suivants:

- la tension ne doit pas dépasser 9V. Il est cependant recommandé de mettre éventuellement les programmes de la carte Raspberry Pi et ceux de la carte A-Star à jour si ceux-ci sont basés sur une valeur de tension différente
- la polarité doit être « positif au centre du connecteur »
- l'alimentation doit pouvoir fournir un courant suffisant, de préférence supérieur ou égal à 1A

6.3.3 - Utilisation

Il est fortement déconseillé de faire des expériences de fonctionnement « rotor bloqué » avec une tension d'alimentation du moteur trop élevée. Ce type d'expérience peut générer des courants trop forts qui réduisent la durée des vies des éléments.

7 - Expériences de commande de moteur à courant continu

Cet ensemble permet de réaliser différentes expériences de commande de moteur à courant continu.

7.1 - Caractéristiques du moteur à courant continu avec codeur incrémental associé

L'expérience de commande de moteur électrique embarque un moteur à courant continu 6V, de rapport de réduction 34:1, avec codeur incrémental 48 CPR (Counts Per Revolution).

Ce moteur est le même que celui utilisé sur le robot Geeros (<http://boutique.3sigma.fr/12-robots>).

Ses équations sont les suivantes:

$$\frac{d}{dt} \omega_m(t) = \frac{\text{ratio} K i_m(t) - d \omega_m(t)}{J \cdot \text{ratio}^2}$$

$$\frac{d}{dt} i_m(t) = \frac{V(t) - R i_m(t) - K \cdot \text{ratio} \omega_m(t)}{L}$$

Avec :

- R : résistance électrique interne: 3.0 Ohms
- L : inductance des enroulements: 3.0 mH
- J : moment d'inertie du rotor: 1.10^{-6} kg.m²
- K : constante de couple = constante de fem: 0.01 N.m/A
- d : coefficient de frottement visqueux: 0.0025 N.m.s/rad
- ω_m : vitesse de rotation de l'arbre de sortie du réducteur (rad/s)
- i_m : courant dans le moteur (A)
- V : tension d'alimentation (V)

Ces paramètres ont été identifiés à partir d'un essai de réponse du moteur à un échelon de tension.

La fonction de transfert entre l'entrée $V(t)$ et la sortie $\omega_m(t)$ est la suivante :

$$\frac{K \text{ratio}}{JL \text{ratio}^2 s^2 + (JR \text{ratio}^2 + Ld) s + K^2 \text{ratio}^2 + Rd}$$

La fonction de transfert entre l'entrée $V(t)$ et la sortie $i_m(t)$ est la suivante :

$$\frac{J \text{ratio}^2 s + d}{JL \text{ratio}^2 s^2 + (JR \text{ratio}^2 + Ld) s + K^2 \text{ratio}^2 + Rd}$$

Noter que les valeurs numériques données ci-dessus concernent l'identification du moteur avec son driver, l'influence de ce dernier étant importante et devant être pris en compte dans l'asservissement global du système.

Le brochage du moteur (couleur des fils) est donné dans le tableau ci-dessous:

Rouge	Noir	Bleu	Vert	Jaune	Blanc
+ moteur	- moteur	5V codeur	Masse codeur	Voie A codeur	Voie B codeur

ATTENTION !

Ne pas confondre la couleur des fils du moteur et la couleur des douilles 2mm.

7.1.1 - Calcul de la vitesse de rotation du moteur

Le codeur incrémental fournit deux signaux carrés en quadrature, comme sur la capture ci-dessous :



Ces deux signaux permettent de mesurer à la fois la vitesse et le sens de rotation. La mesure de la vitesse se fait simplement en comptant le nombre d'impulsions pendant un temps fixe. Les données du problème sont les suivantes :

- Le codeur est fixé à l'arbre moteur et non pas à l'arbre de sortie du réducteur (celui utilisé pour l'entraînement). Le rapport de réduction étant 34:1, l'arbre moteur fait 34 tours lorsque l'arbre du réducteur en fait 1
- Le codeur génère 48 impulsions à chaque fois qu'il fait un tour
- La cadence d'échantillonnage utilisée pour l'asservissement sera de 0.01 s

Par conséquent, lorsque l'arbre principal fait un tour, le codeur génère :
 $34 * 48 \approx 1632$ impulsions.

Si N est le nombre d'impulsions comptées en 0.01 s, la vitesse est (en rad/s, l'unité standard, sachant qu'un tour fait $2*\pi$ radians) :

$$2*\pi*N/(0.01*1632)$$

ATTENTION !

Bien que le codeur soit placé sur l'arbre moteur, le calcul ci-dessus donne la vitesse en sortie du réducteur.

Un point très important concerne la résolution de la mesure, c'est-à-dire la plus petite valeur qu'il est possible de calculer. La formule est la suivante (en rad/s) :

$$2*\pi/(Ts*CPR*ratio)$$

avec :

- Ts : cadence d'échantillonnage
- CPR : nombre d'impulsions par tour du codeur
- ratio : rapport de réduction du moteur

Dans notre cas de figure, la résolution est la suivante

$$2*\pi/(0.01*1632) = 0.385 \text{ rad/s}$$

7.1.2 - Comptage du nombre d'impulsions

Compter le nombre d'impulsions du codeur revient à compter le nombre de fronts montants et descendants des signaux jaune et bleu représentés sur l'image ci-dessus. Pour ce faire, la seule méthode viable consiste à brancher les deux signaux (les fils jaune et blanc sur le codeur utilisé) sur deux entrées « interruption » de la carte A-Star. Les deux autres fils (bleu et vert) seront respectivement branchés sur le 5 V et sur la masse de cette carte.

Sur une carte A-Star, on utilise les lignes d'interruption des broches digitales 0 et 1. L'intérêt d'une ligne d'interruption est qu'elle permet, comme son nom l'indique, d'interrompre le déroulement des calculs sur le processeur pour effectuer un traitement spécifique, en l'occurrence la mise à jour du compteur d'impulsions, avant de rendre la main à la boucle principale.

La seule « difficulté » est de savoir s'il faut incrémenter ou décrémenter le compteur dans le traitement de l'interruption. Il suffit pour cela d'observer les courbes ci-dessus, obtenues alors que le moteur tourne dans le sens positif. On constate que:

- Lorsque la voie A (en jaune) passe au niveau haut, la voie B (en bleu) est au niveau bas
- Lorsque la voie A passe au niveau bas, la voie B est au niveau haut

Quand le moteur tourne dans le sens positif, lors d'une interruption sur la voie A, les niveaux de A et B sont donc inversés.

En ce qui concerne l'interruption liée à la voie B, c'est l'inverse :

- Lorsque la voie B passe au niveau haut, la voie A est au niveau haut
- Lorsque la voie B passe au niveau bas, la voie A est au niveau bas

7.2 - Commande en tension avec la carte Raspberry Pi

Cette expérience permet de changer la vitesse de rotation du moteur en appliquant une tension variable par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Python gérant cette expérience se nomme `CommandeEnTension.py`. Il est situé dans le répertoire `/root/programmes_python`.

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

La carte A-Star doit être programmée avec le programme Arduino dont le nom est de la forme `FirmwareAStar_x.y.zip` (x.y correspond au numéro de version du programme). Il peut être téléchargé à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

Attention : ce programme est préchargé à la livraison du système. Si vous n'avez jamais changé le programme original de la carte A-Star, cette opération n'est pas nécessaire.

Le principe de pilotage du moteur consiste à envoyer des signaux PWM sur la carte A-Star connectée sur la carte Raspberry Pi afin de faire varier la tension d'alimentation du moteur. Cette tension peut être modifiée interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

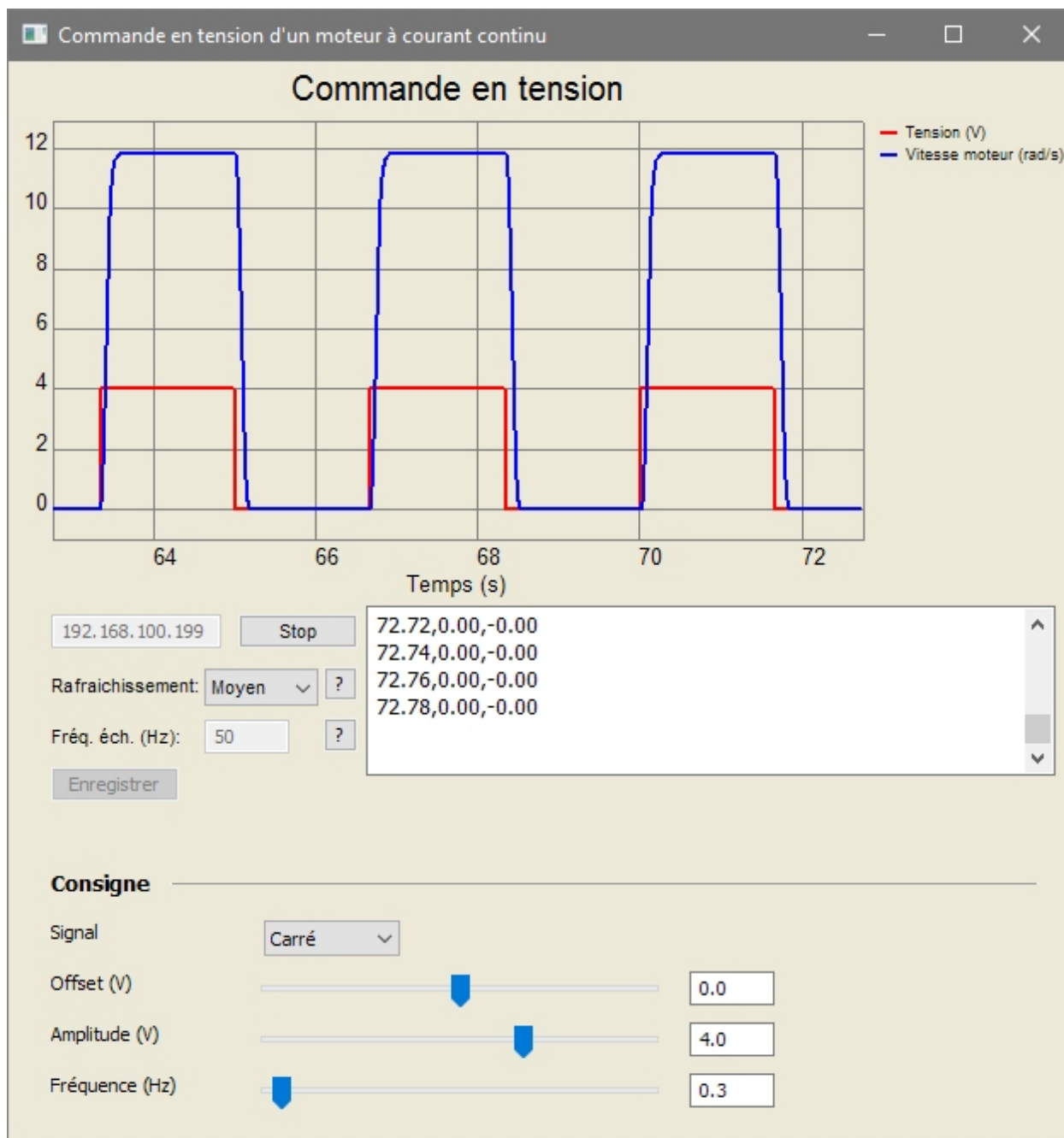
Son nom est de la forme `RaspberryPiCommandeMoteurEnTension_x.y.zip` (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur `RaspberryPiCommandeMoteurEnTension.exe`.

Pour piloter la tension d'alimentation (et donc la vitesse, mais sans asservissement de cette dernière) d'un moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes :

- brancher l'alimentation 9V fournie sur le connecteur jack du système
- connecter l'ordinateur à la carte Raspberry Pi en utilisant le câble Ethernet et l'adaptateur USB – Ethernet (USB côté ordinateur)
- commuter l'interrupteur sur la position « I »
- attendre environ une minute le démarrage de la carte Raspberry Pi
- lancer l'application « `RaspberryPiCommandeMoteurEnTension` »
- entrer l'adresse IP du système (192.168.100.199) dans la zone correspondante et cliquer sur « Connexion ». L'application démarre alors à distance le programme Python de commande du moteur sur la carte Raspberry Pi. Vous pouvez alors interagir avec l'interface graphique pour faire vos expérimentations

Voici une capture d'écran de l'interface de l'application de pilotage:



Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la tension de commande (V, en rouge)
 - la vitesse mesurée (rad/s, en bleu)
- zone de sélection de l'adresse IP: entrer l'adresse IP de votre système et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre en jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafraîchissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme `CommandeEnTension.py` (20 ms par défaut)
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- Zone « Consigne »: elle permet de modifier la consigne de tension avec les curseurs. La vitesse du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel

IMPORTANT !

Si vous comparez la vitesse de rotation obtenue sur votre ensemble avec celle affichée sur la capture d'écran au début de ce paragraphe, vous obtiendrez probablement une valeur différente, même si la tension de commande est la même : c'est normal, ceci est dû à la disparité des caractéristiques des moteurs à courant continu.

D'un point de vue pédagogique, ce point permet de souligner la nécessité de réaliser un asservissement de vitesse : si l'on souhaite une vitesse de rotation précise, on ne peut pas se contenter d'une commande en boucle ouverte.

7.3 - Asservissement de vitesse avec la carte Raspberry Pi

Cette expérience permet d'asservir la vitesse de rotation du moteur en appliquant une consigne de vitesse par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Python gérant cette expérience se nomme AsservissementVitesse.py. Il est situé dans le répertoire /root/programmes_python.

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

La carte A-Star doit être programmée avec le programme Arduino dont le nom est de la forme FirmwareAStar_x.y.zip (x.y correspond au numéro de version du programme). Il peut être téléchargé à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

Attention : ce programme est préchargé à la livraison du système. Si vous n'avez jamais changé le programme original de la carte A-Star, cette opération n'est pas nécessaire.

Le principe de pilotage du moteur consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer au moteur (via commande PWM de la carte A-Star connectée sur la carte Raspberry Pi) pour qu'il suive la consigne de vitesse spécifiée.

Notez que dans ce programme, la vitesse mesurée est en fait la moyenne glissante des 10 derniers échantillons de mesure de vitesse instantanée, ce qui permet d'avoir une mesure plus lisse. En effet, la résolution de la mesure instantanée est la suivante:

$$2*\pi/(Ts*CPR*ratio)$$

avec :

- Ts : cadence d'échantillonnage (0.01 s)
- CPR : nombre d'impulsions par tour du codeur (48)
- ratio : rapport de réduction du moteur (34)

Ceci donne une résolution de $2*\pi/(0.01*1632) = 0.385$ rad/s. Le moyennage permet d'améliorer cette valeur.

La consigne de vitesse peut se fixer interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

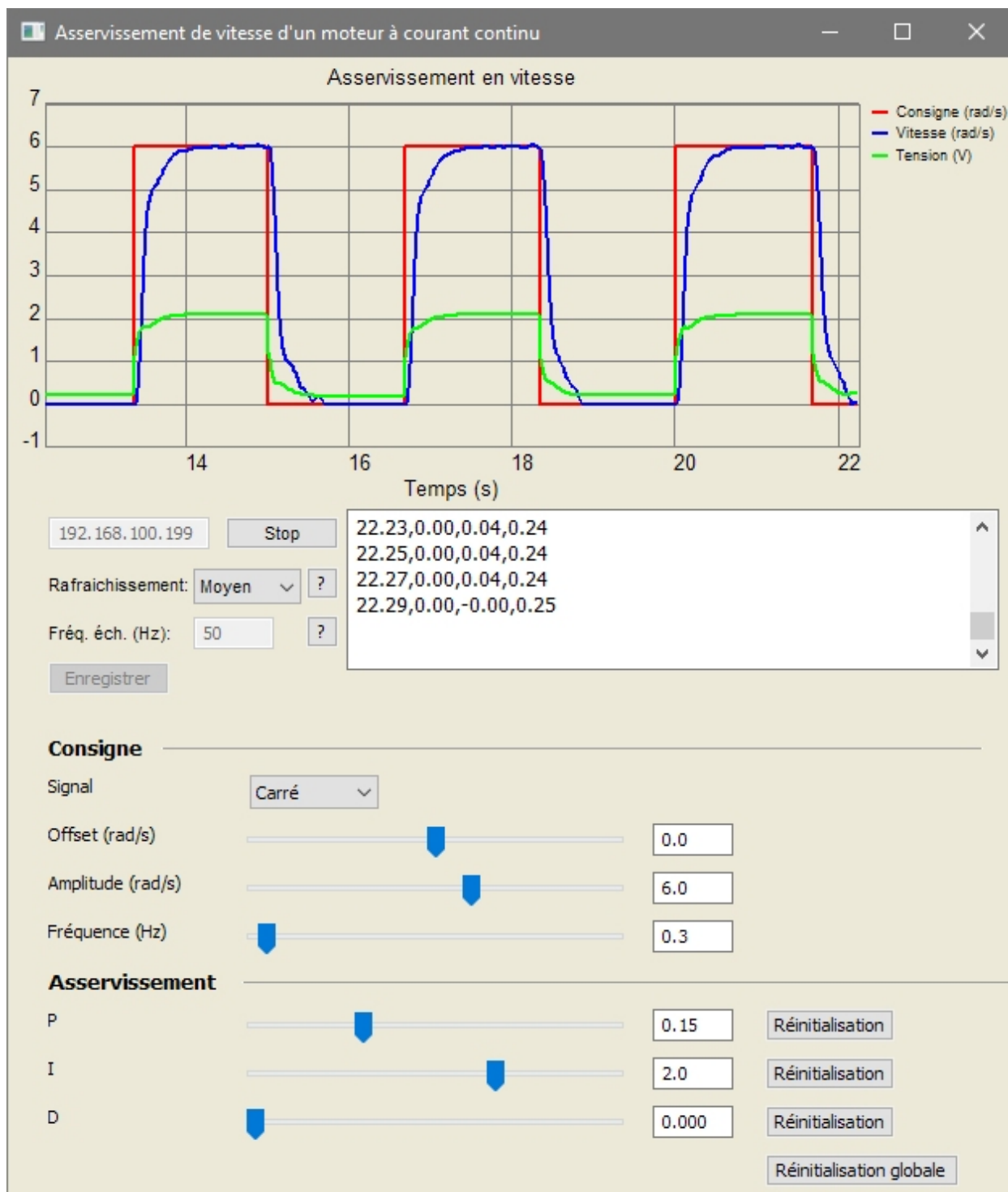
Son nom est de la forme RaspberryPiAsservissementMoteurEnVitesse_x.y.zip (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur RaspberryPiAsservissementMoteurEnVitesse.exe.

Pour changer la consigne de vitesse du moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes (remarque: n'exécutez pas de nouveau celles que vous avez déjà effectuées):

- brancher l'alimentation 9V fournie sur le connecteur jack du système
- connecter l'ordinateur à la carte Raspberry Pi en utilisant le câble Ethernet et l'adaptateur USB – Ethernet (USB côté ordinateur)
- commuter l'interrupteur sur la position « I »
- attendre environ une minute le démarrage de la carte Raspberry Pi
- lancer l'application « RaspberryPiAsservissementMoteurEnVitesse »
- entrer l'adresse IP du système (192.168.100.199) dans la zone correspondante et cliquer sur « Connexion ». L'application démarre alors à distance le programme Python d'asservissement du moteur sur la carte Raspberry Pi. Vous pouvez alors interagir avec l'interface graphique pour faire vos expérimentations

Voici une capture d'écran de l'interface de l'application de pilotage:



Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la consigne de vitesse (rad/s, en rouge)
 - la vitesse mesurée (rad/s, en bleu)
 - la tension de commande (V, en vert)
- zone de sélection de l'adresse IP: entrer l'adresse IP de votre système et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre en jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafraîchissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme AsservissementVitesse.py.
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- Zone « Consigne »: elle permet de modifier la consigne de vitesse avec les curseurs. La vitesse du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel
- Zone « Asservissement »: elle permet de modifier les gains du régulateur PID pendant le fonctionnement du système. Cela permet de voir l'influence de ces gains sur les performances de l'asservissement et de régler précisément ce dernier.

7.4 - Asservissement de position avec la carte Raspberry Pi

Cette expérience permet d'asservir la position de rotation du moteur en appliquant une consigne de position par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Python gérant cette expérience se nomme AsservissementPosition.py. Il est situé dans le répertoire /root/programmes_python.

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

La carte A-Star doit être programmée avec le programme Arduino dont le nom est de la forme FirmwareAStar_x.y.zip (x.y correspond au numéro de version du programme). Il peut être téléchargé à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

Attention : ce programme est préchargé à la livraison du système. Si vous n'avez jamais changé le programme original de la carte A-Star, cette opération n'est pas nécessaire.

Le principe de pilotage du moteur consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer au moteur (via commande PWM de la carte A-Star connectée sur la carte Raspberry Pi) pour qu'il suive la consigne de position spécifiée.

Notez que dans ce programme, la position mesurée est en fait la moyenne glissante des 10 derniers échantillons de mesure de position instantanée, ce qui permet d'avoir une mesure plus lisse. En effet, la résolution de la mesure instantanée est la suivante:

$2\pi / (\text{CPR} * \text{ratio})$

avec :

- CPR : nombre d'impulsions par tour du codeur (48)
- ratio : rapport de réduction du moteur (34)

Ceci donne une résolution de $2\pi / 1632 = 0.00385$ rad. Le moyennage permet d'améliorer cette valeur.

La consigne de position peut se fixer interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

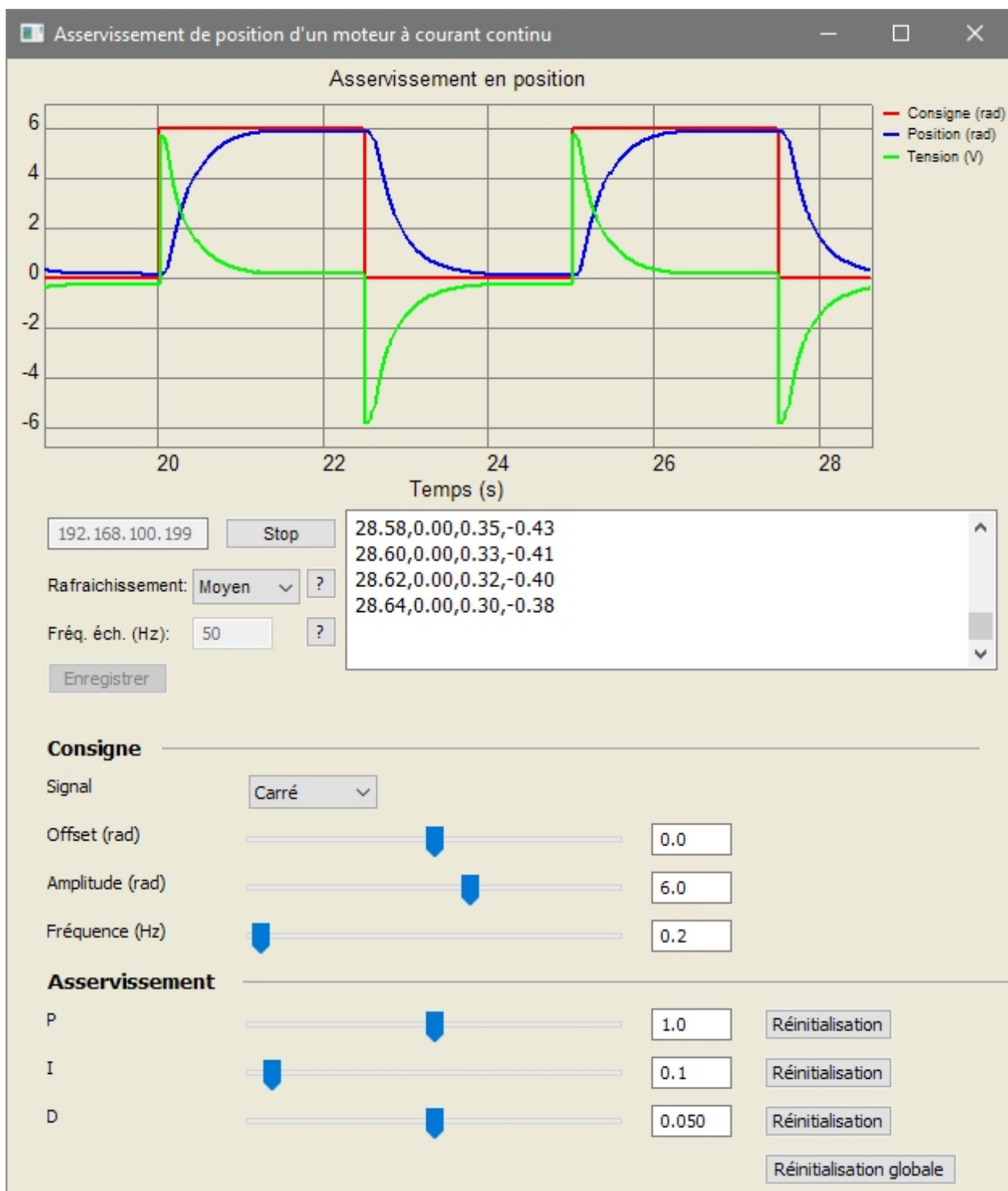
Son nom est de la forme RaspberryPiAsservissementMoteurEnPosition_x.y.zip (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur RaspberryPiAsservissementMoteurEnPosition.exe.

Pour changer la consigne de position du moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes (remarque: n'exécutez pas de nouveau celles que vous avez déjà effectuées):

- brancher l'alimentation 9V fournie sur le connecteur jack du système
- connecter l'ordinateur à la carte Raspberry Pi en utilisant le câble Ethernet et l'adaptateur USB – Ethernet (USB côté ordinateur)
- commuter l'interrupteur sur la position « I »
- attendre environ une minute le démarrage de la carte Raspberry Pi
- lancer l'application « RaspberryPiAsservissementMoteurEnPosition »
- entrer l'adresse IP du système (192.168.100.199) dans la zone correspondante et cliquer sur « Connexion ». L'application démarre alors à distance le programme Python d'asservissement du moteur sur la carte Raspberry Pi. Vous pouvez alors interagir avec l'interface graphique pour faire vos expérimentations

Voici une capture d'écran de l'interface de l'application de pilotage:



Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la consigne de position (rad, en rouge)
 - la position mesurée (rad, en bleu)
 - la tension de commande (V, en vert)
- zone de sélection de l'adresse IP: entrer l'adresse IP de votre système et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre en jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafraîchissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme AsservissementVitesse.py.
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- Zone « Consigne »: elle permet de modifier la consigne de position avec les curseurs. La position du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel
- Zone « Asservissement »: elle permet de modifier les gains du régulateur PID pendant le fonctionnement du système. Cela permet de voir l'influence de ces gains sur les performances de l'asservissement et de régler précisément ce dernier.

7.5 - Commande en tension avec la carte A-Star

Cette expérience permet de changer la vitesse de rotation du moteur en appliquant une tension variable par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Arduino peut être téléchargé à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

Son nom est de la forme AStarCommandeEnTension_x.y.zip (x.y correspond au numéro de version du programme).

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

Le principe de pilotage du moteur consiste à envoyer des signaux PWM sur le pont en H intégré à la carte A-Star afin de faire varier la tension d'alimentation du moteur. Cette tension peut être modifiée interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

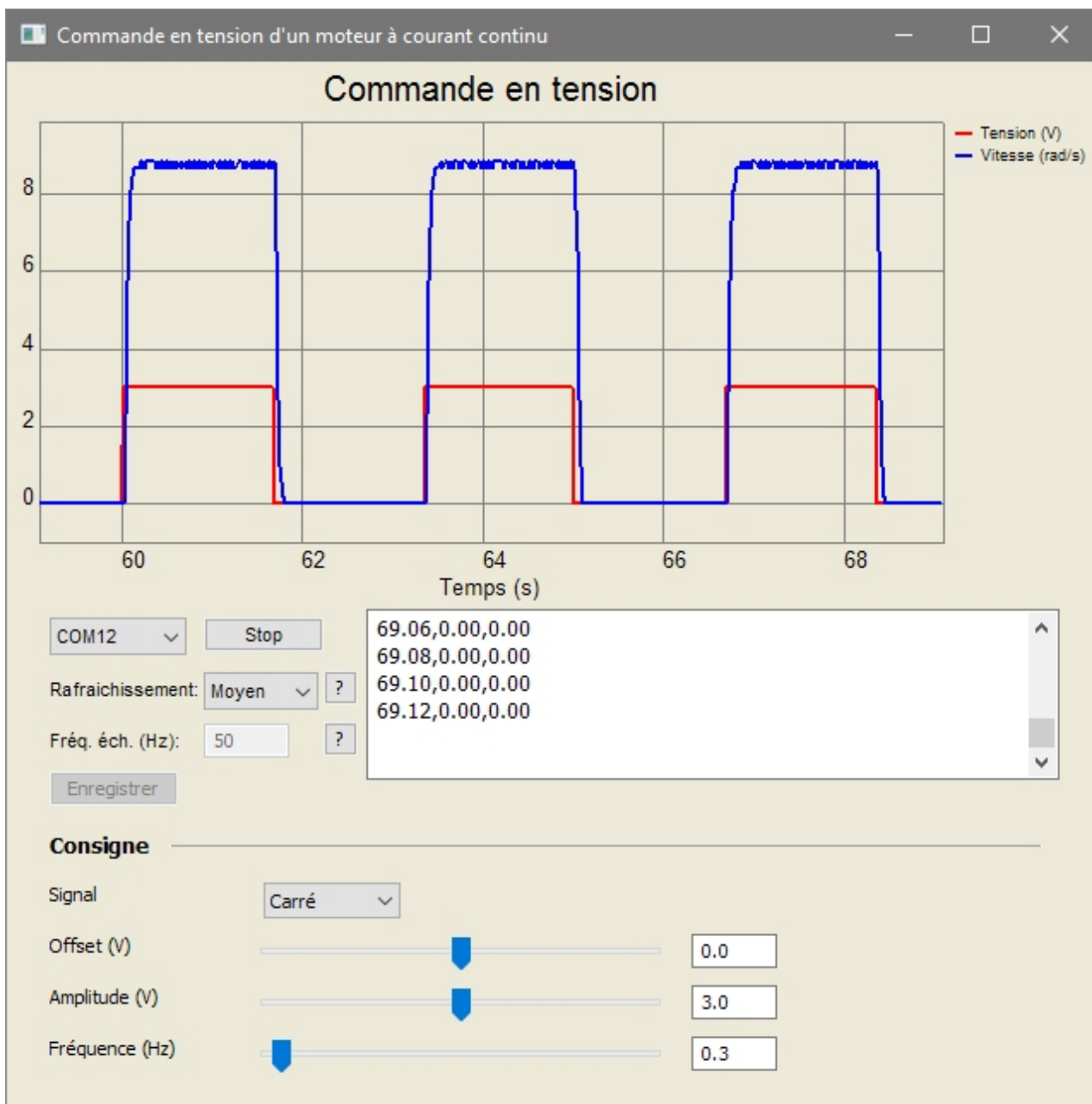
Son nom est de la forme AStarCommandeMoteurEnTension_x.y.zip (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur AStarCommandeMoteurEnTension_x.y.exe.

Pour piloter la tension d'alimentation (et donc la vitesse) du moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes (remarque: n'exécutez pas de nouveau celles que vous avez déjà effectuées):

- télécharger l'application de pilotage « AStarCommandeMoteurEnTension » à l'adresse ci-dessus et l'installer
- télécharger et installer l'IDE Arduino
- connecter l'alimentation 9V sur le connecteur jack du système
- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I »
- connecter la carte AStar à l'ordinateur avec le câble USB fourni
- lancer l'application « AStarCommandeMoteurEnTension »

Voici une capture d'écran de l'interface de l'application de pilotage:



Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la tension de commande (V, en rouge)
 - la vitesse mesurée (rad/s, en bleu)
- zone de sélection du port série: choisir le port série sur lequel est connectée votre carte AStar et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre à jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafrâichissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme Arduino. Attention: ce paramètre doit être spécifié en Hz, alors que les valeurs correspondantes dans le programme Arduino (CADENCE_MS et TSDATA) sont spécifiées en ms. La relation entre les deux est la suivante:
$$\text{freq (Hz)} = 1000 / \text{cadence (ms)}$$
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- Zone « Consigne »: elle permet de modifier la consigne de tension avec les curseurs. La vitesse du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel

IMPORTANT !

Si l'affichage ne suit pas les consignes même avec une fréquence de rafraîchissement très lente, veuillez diminuer la cadence d'échantillonnage dans le programme Arduino (augmenter les valeurs TSDATA et CADENCE_MS)

IMPORTANT !

Si vous comparez la vitesse de rotation obtenue sur votre ensemble avec celle affichée sur la capture d'écran au début de ce paragraphe, vous obtiendrez probablement une valeur différente, même si la tension de commande est la même : c'est normal, ceci est dû à la disparité des caractéristiques des moteurs à courant continu.

D'un point de vue pédagogique, ce point permet de souligner la nécessité de réaliser un asservissement de vitesse : si l'on souhaite une vitesse de rotation précise, on ne peut pas se contenter d'une commande en boucle ouverte.

7.6 - Asservissement de vitesse avec la carte A-Star

Cette expérience permet d'asservir la vitesse de rotation du moteur en appliquant une consigne de vitesse par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Arduino peut être téléchargé à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

Son nom est de la forme AStarAsservissementVitesse_x.y.zip (x.y correspond au numéro de version du programme).

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

Le principe de pilotage du moteur consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer au moteur (via commande PWM du pont en H intégré sur la carte AStar) pour qu'il suive la consigne de vitesse spécifiée.

Notez que dans ce programme, la vitesse mesurée est en fait la moyenne glissante des 10 derniers échantillons de mesure de vitesse instantanée, ce qui permet d'avoir une mesure plus lisse. En effet, la résolution de la mesure instantanée est la suivante:

$$2*\pi/(Ts*CPR*ratio)$$

avec :

- Ts : cadence d'échantillonnage (0.01 s)
- CPR : nombre d'impulsions par tour du codeur (48)
- ratio : rapport de réduction du moteur (34)

Ceci donne une résolution de $2*\pi/(0.01*1632) = 0.385$ rad/s. Le moyennage permet d'améliorer cette valeur.

La consigne de vitesse peut se fixer interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

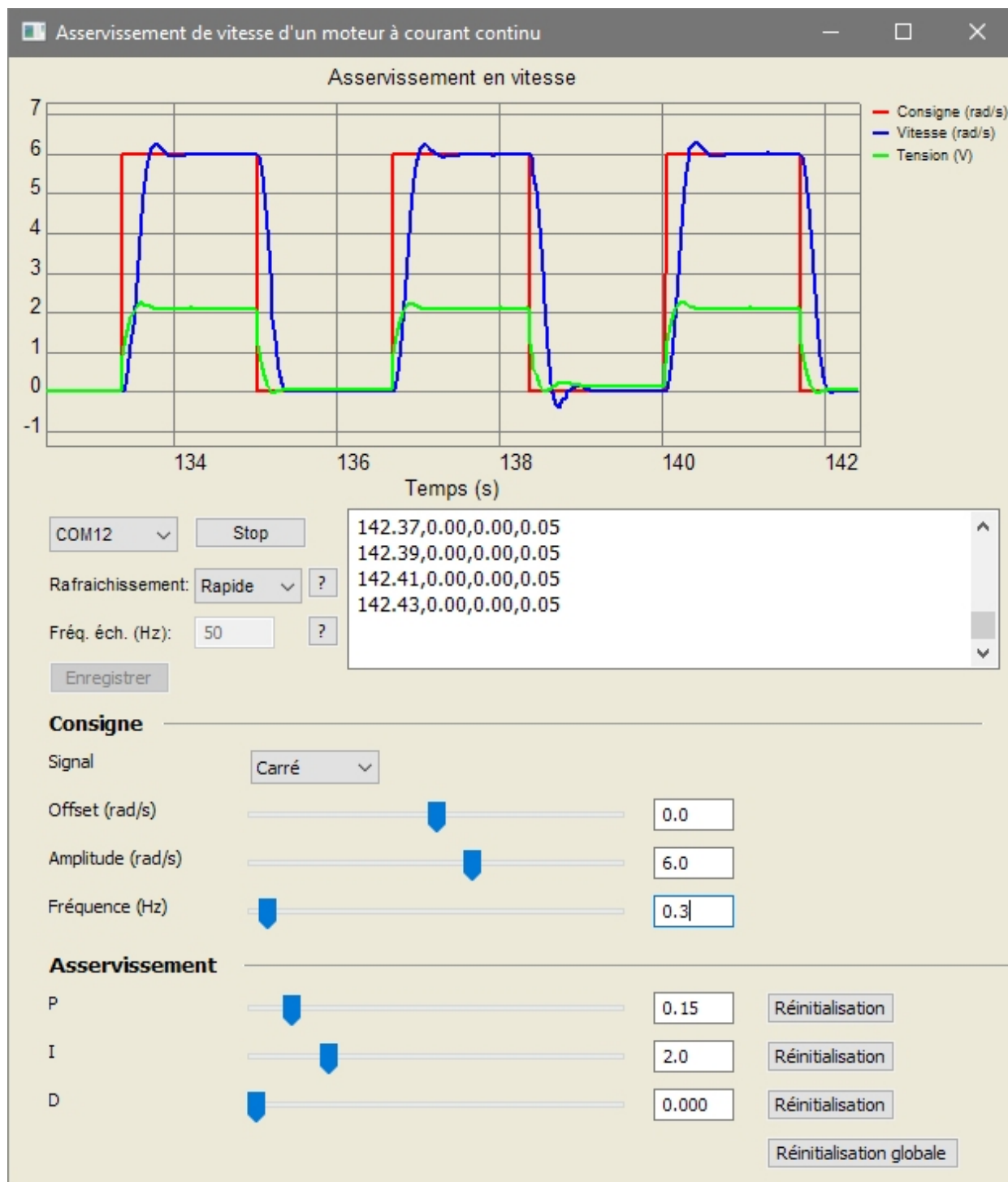
Son nom est de la forme AStarAsservissementMoteurEnVitesse_x.y.zip (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur AStarAsservissementMoteurEnVitesse_x.y.exe.

Pour changer la consigne de vitesse du moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes (remarque: n'exécutez pas de nouveau celles que vous avez déjà effectuées):

- télécharger l'application de pilotage «AStarAsservissementMoteurEnVitesse» à l'adresse ci-dessus et l'installer
- télécharger et installer l'IDE Arduino
- connecter l'alimentation 9V sur le connecteur jack du système
- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I »
- connecter la carte AStar à l'ordinateur avec le câble USB fourni
- lancer l'application «AStarAsservissementMoteurEnVitesse»

Voici une capture d'écran de l'interface de l'application de pilotage:



Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la consigne de vitesse (rad/s, en rouge)
 - la vitesse mesurée (rad/s, en bleu)
 - la tension de commande (V, en vert)
- zone de sélection du port série: choisir le port série sur lequel est connectée votre carte AStar et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre à jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafraîchissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme Arduino. Attention: ce paramètre doit être spécifié en Hz, alors que la valeur correspondante dans le programme Arduino (TSDATA) est spécifiée en ms. La relation entre les deux est la suivante:
$$\text{freq (Hz)} = 1000 / \text{TSDATA (ms)}$$
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- Zone « Consigne »: elle permet de modifier la consigne de vitesse avec les curseurs. La vitesse du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel
- Zone « Asservissement »: elle permet de modifier les gains du régulateur PID pendant le fonctionnement du système. Cela permet de voir l'influence de ces gains sur les performances de l'asservissement et de régler précisément ce dernier.

IMPORTANT !

Si l'affichage ne suit pas les consignes même avec une fréquence de rafraîchissement très lente, veuillez diminuer la cadence d'échantillonnage dans le programme Arduino (augmenter la valeur TSDATA). Ne pas modifier la valeur CADENCE_MS, sous peine de déstabiliser l'asservissement.

7.7 - Asservissement de position avec la carte A-Star

Cette expérience permet d'asservir la position angulaire du moteur en appliquant une consigne d'angle par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Arduino peut être téléchargé à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

Son nom est de la forme AStarAsservissementPosition_x.y.zip (x.y correspond au numéro de version du programme).

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

Le principe de pilotage du moteur consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer au moteur (via commande PWM du pont en H intégré sur la carte AStar) pour qu'il suive la consigne de position spécifiée.

La consigne de position peut se fixer interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

http://www.3sigma.fr/Telechargements-Ensemble_Commande_de_moteur_a_courant_continu.html

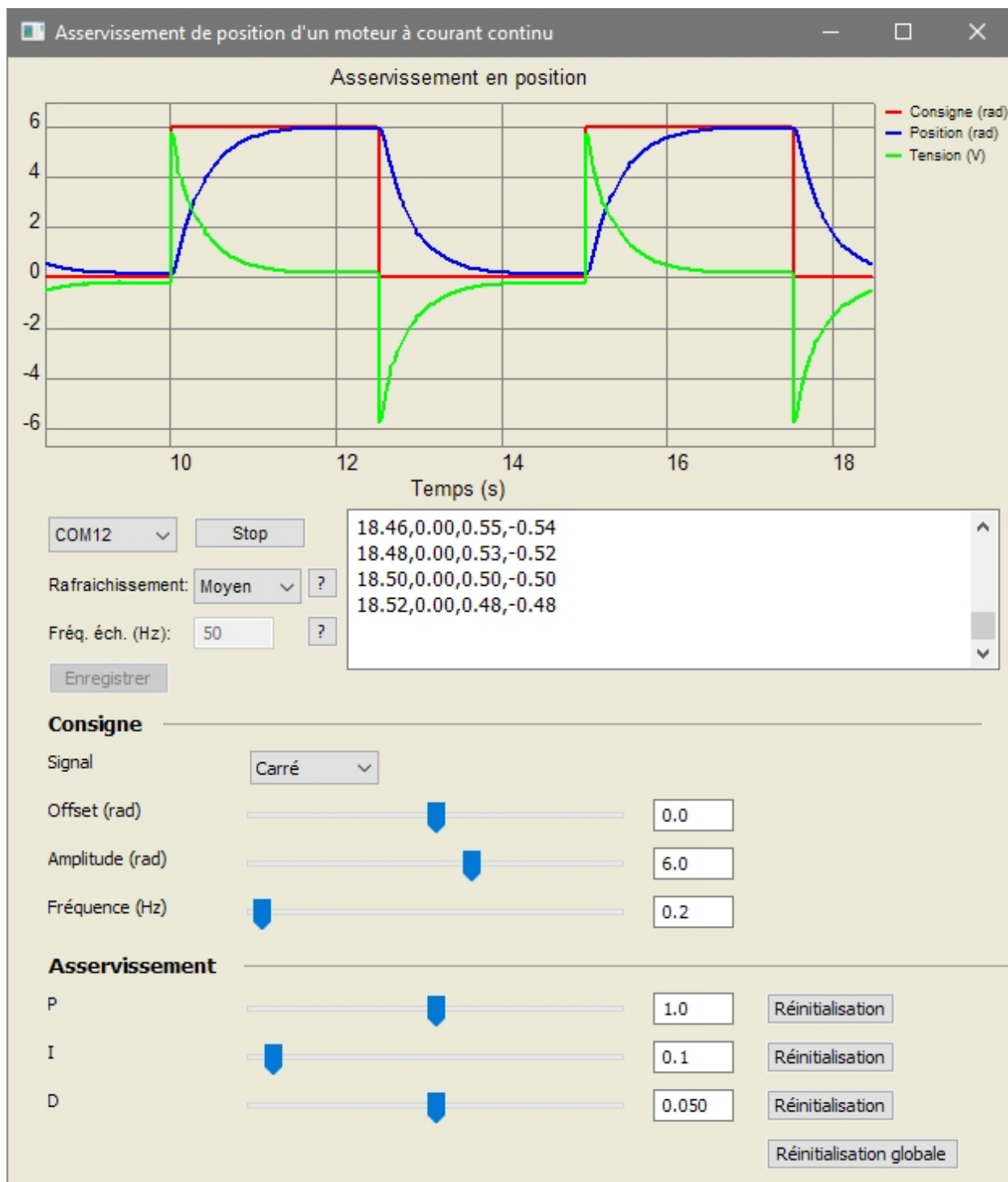
Son nom est de la forme AStarAsservissementMoteurEnPosition_x.y.zip (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur AStarAsservissementMoteurEnPosition_x.y.exe.

Pour changer la consigne de position du moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes (remarque: n'exécutez pas de nouveau celles que vous avez déjà effectuées):

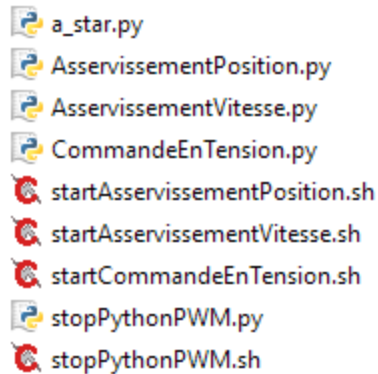
- télécharger l'application de pilotage «AStarAsservissementMoteurEnPosition» à l'adresse ci-dessus et l'installer
- télécharger et installer l'IDE Arduino
- connecter l'alimentation 9V sur le connecteur jack du système
- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I »
- connecter la carte AStar à l'ordinateur avec le câble USB fourni
- lancer l'application «AStarAsservissementMoteurEnPosition»

Voici une capture d'écran de l'interface de l'application de pilotage:



8 - Accès aux programmes Python

Vous pouvez accéder aux programmes Python de gestion du moteur sur la carte Raspberry Pi dans le répertoire `/root/programmes_python` :



Ces programmes sont librement consultables et modifiables en vous connectant en SSH à la carte Raspberry Pi avec les informations suivantes :

- adresse IP : 192.168.100.199
- login : root
- mot de passe : raspberrypi

Nous vous recommandons l'utilisation du logiciel WinSCP (<https://winscp.net/eng/download.php>) qui permet de naviguer très aisément dans l'arborescence des répertoires de la carte Raspberry Pi et de faire facilement des transferts de fichiers entre votre ordinateur hôte et la Raspberry Pi.

9 - Important

Cet ensemble est un produit « vivant » en constant développement pour améliorer ou lui ajouter de nouvelles fonctionnalités. Si vous avez des idées ou des besoins pour des développements spécifiques, n'hésitez pas à nous contacter (info@3sigma.fr).

Ne restez jamais bloqué sans nous contacter !

Pour tout problème ou toute requête, contactez-nous à l'adresse support@3sigma.fr.